A framework for aligning and indexing movies with their script

Rémi Ronfard, Tien Tran-Thuong

A FRAMEWORK FOR ALIGNING AND INDEXING MOVIES WITH THEIR SCRIPT

Remi Ronfard and Tien Tran Thuong

INRIA Rhone Alpes
Montbonnot, France

ABSTRACT

A continuity script describes very carefully the content of a movie shot by shot. This paper introduces a framework for extracting structural units such as shots, scenes, actions and dialogs from the script, and aligning them to the movie based on the *longest matching subsequence* between them. We present experimental results and applications of the framework with a full-length movie and discuss its applicability to large-scale film repositories.

1. INTRODUCTION

Choosing terms for describing and indexing video content is a difficult and important problem. We believe not enough attention has been given to a very important source of video descriptions - the *continuity script* which describes very carefully the content of a movie shot by shot. In this paper, we discuss some of the issues related with synchronizing and aligning a movie with its script using a combination of cues from the dialogs and the image track. We describe grammars and automata for formatting the script into structural units such as shots, scenes, actions and dialogs. We then introduce a dynamic programming algorithm for finding the longest matching subsequence between the formatted script and the video content. This procedure aligns the script to the temporal axis of the movie at the shot and dialog levels, and therefore allows dialogs and action descriptions in the script to be used as indices to the video content. We illustrate the framework with 'The wizard of Oz ', a well-known masterpiece released in 1939, whose continuity script was carefully edited and published on the Internet [1].

Alignment of script to video was mentioned by other researchers [2, 3, 4] as a means to provide training data for learning models of objects, scenes and actors. But contrary to the similar problem of aligning bilingual translations of the same text [5, 6], it was never formalized properly. With this work, we would like to contribute to such a formalization.

2. SCRIPT FORMATTING

In this section, we introduce our model of the continuity script for 'The wizard of Oz', and algorithms for automatically formatting the script from plain text to XML. Typically, a continuity script is updated throughout shooting of the movie and includes the breakdown of scenes into shots. In contrast, a production script only breaks down the movie into its master scenes. In that case, the alignment and indexing can only be performed at a much grosser scale. In this work, we are particularly interested in continuity scripts, such as 'The wizard of Oz'.

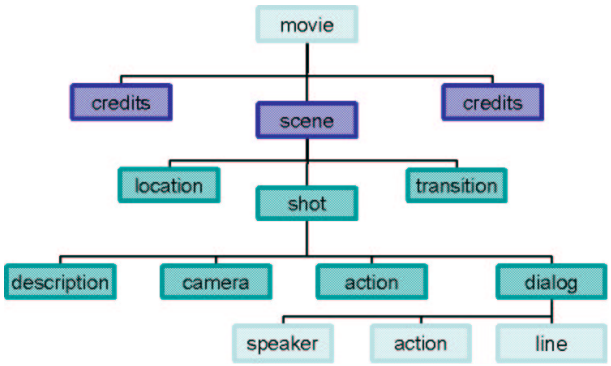

Fig. 1. *High-level grammar of film structure. A movie is composed of scenes, which are composed of shots. Transitions can occur between shots or scenes. Shots are composed of actions, camera movements and dialogs.*

From our own analysis of many film scripts and related books in film studies, we found that the structural components of a film script were - the scene, the shot, the transition, the action, the camera action and the dialog, as represented in Fig. 1. *Scene* is a segment of the movie taking place in a given location. It is described as 'interior' or 'exterior' and with the name of a place or location. It contains a sequence of contiguous shots. *Shot* has type close-shot (CS), medium-close-shot (MCS), medium shot (MS), medium-long-shot (MLS), long-shot (LS) or extreme-long-shot (ELS). It starts with a description, usually naming the actors, the settings and the camera viewpoint, followed by a

sequence of actions, camera actions and dialogs. *Transition* has type dissolve or fade and separates two shots or scenes. *Camera* is an informal textual description of the camera motion. *Action* is an informal textual description of an action taking place within a shot. It usually names the action with a verb as well the actors performing the action, and includes references to places in the scene and on the screen. *Dialog* starts with the name of a speaker. It contains a sequence of utterances (broken into lines) and actions.

The organization of those components in a particular script is embodied by a set of typographic and stylistic rules. In order to format the script into a strict, structured representation, we need to further describe those rules as a grammar, down to terminal symbols such as letters, tabulations and line breaks. It turns out that in many classical Hollywood-style scripts, the grammar is regular. In other words, film scripts can be modelled with regular expressions and recognized with finite-state automata. As an example, the formatting rules for the continuity script of the 'Wizard of Oz' follows the grammar of Fig 2.

SCENARIO	→	CREDITS? SCENE + CREDITS?
SCENE	→	LOCATION (SHOT[TRANSITION]) +
LOCATION	→	("INT." | "EXT.") – TEXT
TRANSITION	→	TAB? ("FADE IN" | "FADE OUT"
TRANSITION	→	TAB? "LAP DISSOLVE TO")
SHOT	→	SIZE DESCRIPTION DIALOG +
SIZE	→	"CS" | "MCS" | "MS" | "MLS" | "LS"
DESCRIPTION	→	ACTION [–ACTION | CAMERA]+
DIALOG	→	TAB SPEAKER "(O.S.)"? [LINE | ACTION]+
CAMERA	→	"CAMERA" TEXT
ACTION	→	TEXT

Fig. 2. *A grammar for the continuity script for 'The Wizard of Oz'. Higher-level symbols from Fig. 1 are explicitly decomposed into lower-level entities and terminals (formatting and tabulations). This grammar is easily found to be regular since all productions are either of type $A \rightarrow a$ or $A \rightarrow aB$, where a is a terminal and A, B are non-terminals.*

As a result, the script can be analyzed as a regular expression with a finite state automaton. Once this grammar has been fully worked out, it is easy to write down an automaton for transcribing the entire script into an XML tree in the form of Fig. 1. Fig. 3 shows the three shots of Fig. 4 translated into XML using specialized tags for shots, actions, cameras and dialogs.

3. SCRIPT ALIGNMENT

Given the formatted script, we now have to align its elements with the temporal axis of the movie, so that the descriptions from the script can be used as indices to the video content. This is not a trivial task because the video comes as large chunk of data, which must be parsed into elements corresponding to the scenes, shots, actions and dialogs in

```
<shot size="CS">
<action>Toto by wheel of rake</action>
<action>listening to song</action>
</shot>
<shot size="MCS">
<action>Dorothy singing</action>
<action>swings on wheel of rake</action>
<action>then walks forward around wheel</action>
<action>Toto jumps up onto seat of rake</action>
<action>Dorothy pets him</action>
<camera>CAMERA PULLS back</camera>
<dialog speaker="DOROTHY">
<line>Someday I'll wish upon a star</line>
</dialog>
</shot>
<shot size="LS">
<action>Miss Gulch rides forward</action>
<action>stops and gets off her bicycle</action>
</shot>
```

Fig. 3. *Example of xml-formatted script. For lack of space, we did not reproduce the scene level, where in fact the first two shots are part of the same scene and the third shot introduces a new scene.*

the script. Since there may be errors in both the formatting of the script and the parsing of the video, the alignment should be flexible enough. While video parsing has a long and active history, we do not believe that the results of video analysis can be trusted to generate a full tree structure allowing to formulate the alignment as a tree-matching problem. Instead, we temporally sort all the script elements and extracted video segments (shot transitions and subtitles) and apply string matching techniques to align them.

In this section, we reformulate the alignment problem as one of finding the longest matching subsequences (LMSS) between the movie and the script, and we describe an efficient dynamic programming algorithm which solves this problem. Dynamic programming has been used with much success for aligning bilingual corpora [5] and for matching video sequences [7].

For the purpose of aligning a movie and a script, we extract subtitles and candidate shot cuts. We implemented and used an algorithm described by Salesin et al. for shot change detection using Haar wavelet coefficients [8]. The algorithm computes a distance between successive frames, and uses thresholds to detect candidate shot cuts. We carefully tuned the thresholds over multiple temporal resolutions to obtain quasi perfect precision (no false detection). This results in a fast and reasonably robust detection, except for the case of dissolves and fades, which result in a large number of missed shot transitions. Gradual transitions are still an open issue for many other algorithms, because they typically introduce large numbers of false detections. In the context of

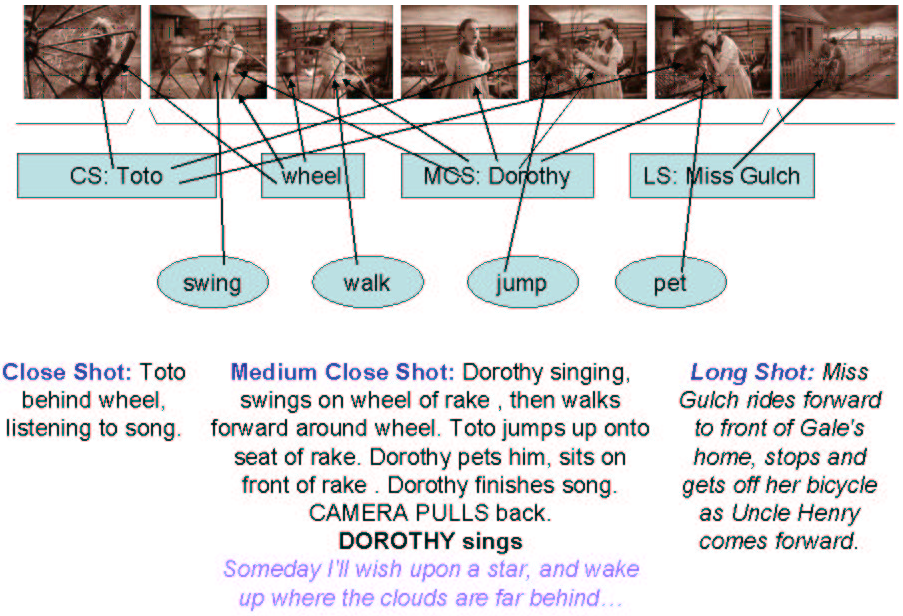

Fig. 4. *Example of three shots aligned to their script descriptions in* The wizard of Oz. *Shots in the script are aligned to automatically detected shot changes in the video. Aligned shots are described by the locations, actors and actions mentioned in the script.*

this work, we were interested to verify the viability of our alignment framework with imperfect shot detection, under the assumption that the effect of missed transitions would remain local (as was effectively verified).

Separately, we extracted the English subtitles from the same video stream and performed optical character recognition on them to produce a stream of time-stamped short texts. The detected shots and subtitles were translated into an MPEG7-like XML format for matching.

The alignment between the script and the detected video segments was performed by matching a temporally sorted string of shots and dialog lines from the script with the shots and subtitles from the video. More specifically, we searched for the longest increasing subsequence of matched shots and dialog/subtitles, a problem which can be solved efficiently with a classic dynamic programming algorithm [9]. In its simplest form, this approach accounts for the following three cases when comparing segments from the script and the movie - either they match, or the script segment was deleted, or the video segment was inserted. Note that this approach can be generalized in many ways to include more sophisticated editing models.

4. INDEXING AND SYNCHRONIZATION

Formatting and synchronizing the movie script for 'The wizard of Oz' opened up two useful and interesting applications, which proved surprisingly easy to implement using XSL transformations on the matched subsequences. In the first application, we created a database of all the scenes, shots, actions and dialogs of the movie and indexed them with the corresponding text from the script. In addition, the formatting of the script allowed us to extract and categorize place/location names (from scene descriptions), speaker/actor names (from dialogs) and action verbs (from shot descriptions).

In the second application, we generated MPEG-7 like elements and their temporal relations for use in an enhanced multimedia player for film studies which we implemented using our MDEFI framework. MDEFI [1] is an advanced environment for playing and editing multimedia documents [10]. MDEFI is based on Madeus, an extension of SMIL with the additional features of (1) enhanced separation of media content location, temporal information and spatial information, (2) hierarchical, operator-based temporal model complemented with relations, (3) rich spatial specification model with relative placements and (4) media fragment integration. MDEFI allows to reformat media content descriptions based on the MPEG-7 standard. This description is then used for specifying fine-grained composition between media objects. In this work, the fine-grained composition features of MDEFI were used to synchronize the video shots and the film script. When playing the movie, for example, the corresponding parts of the film script can be highlighted in synchronization. In addition, the user can jump to video

[1] Multimedia DEscription and Fine-grained Integration

segments by clicking anywhere on the script - as long as a matched segment can be found.

5. EXPERIMENTAL RESULTS

We performed the alignment of 'The wizard of Oz', starting with 791 shots in the script and 683 detected shots. Dissolves and special effects such as explosions and transitions through a crystal ball could not be detected at this stage. The alignment was performed using 2649 subtitles extracted from the video and 3041 dialog lines in the script. We compared dialogs and subtitles using approximate string matching [2], successfully matching a total of 1866 dialog lines. As a result, we were able to automatically align 604 shots, leaving 187 scenario shots unmatched and 79 video shots unmatched. A rapid manual inspection revealed that most of the matched shots were matched correctly, except for a few, highly localized segments of the movies with either (1) a fast succession of *missed* dissolves and special effects or (2) a missing scene, which was edited out from the script in the final movie. The latter case accounts for 80 unmatched shots. Of the remaining 107 unmatched shots in the script, half were due to undetected transitions and half to smaller variations between the final movie and the script. Our alignment algorithm therefore correctly matched 82% of the script shots and 88% of the detected video shots, reducing the number of outstanding shots from 791 to 107.

Of course, future work will be devoted to the remaining fraction of shots and dialogs which could not be matched with our current method. We are following two main directions of research in this respect. On the one hand, we can improve the alignment of shots (especially those without dialogs) by matching visual descriptors in addition to subtitles and compensating for inaccuracies in the shot detection algorithm by matching all frames, using models of the expected shot durations. This would match at the frame, rather than shot level, and use shot transition probabilities, rather than hard decisions in order to handle the more difficult cases of dissolves and special effects better[3]. On the other hand, we are extending our alignment algorithm following previous work in machine translation [5] to account for more elaborate models of insertions, deletions and replacements between the movie and script shots, based on the experimentation reported here. We are also interested in generalizing to other scripts and script formats, which entails discovering the formatting rules for the new scripts, writing down their grammars and checking that they remain consistent. Finally, we believe this work opens the way for even more ambitious developments such as tracking and hyper-linking of video objects and spatio-temporal synchronization, which are already part of the MDEFI framework.

[2] Actually, another instance of the longest common subsequence search!

[3] This will effectively turn our longest matching subsequence algorithm into a Hidden Markov Model decoding algorithm

6. CONCLUSION

By examining the script of 'The wizard of Oz', we have found that *at the structural level* at least, a movie and its script can be analyzed and synchronized with simple tools (regular expressions and dynamic programming). This has allowed us to format the script into high-level components and to align some of them to the movie itself. As a result of this work, we are currently building a large database of movie shots, indexed by dialogs, actors, settings and action descriptions. We believe such a database can be useful for film studies as well as for learning statistical models of video content.

7. REFERENCES

[1] Noel Langley, Florence Ryerson, and Edgar Allen Woolf, "The wizard of oz– movie script," 1939, Cutting Continuity Script, Taken From Printer's Dupe, Last revised March 15, 1939. This script was transcribed by Paul Rudoff.

[2] Joshua S. Wachman and Rosalind W. Picard, "Tools for browsing a TV situation comedy based on content specific attributes," *Multimedia Tools and Applications*, vol. 13, no. 3, pp. 255–284, 2001.

[3] Salway and Tomadaki, "Temporal information in collateral texts for indexing moving images," in *Proceedings of LREC 2002 Workshop on Annotation Standards for Temporal Information in Natural Language*, 2002.

[4] C.G.M. Snoek and M. Worring, "Multimodal video indexing: A review of the state-of-the-art," *Multimedia Tools and Applications*, 2003, Accepted for publication.

[5] W. A. Gale and K. W. Church, "A program for aligning sentences in bilingual corpora," in *In Proceedings of ACL-91, Berkeley CA.*, 1991.

[6] P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer, "The mathematics of machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, 1993.

[7] Milind R. Naphade, Roy Wang, and Thomas S. Huang, "Supporting audiovisual query using dynamic programming," in *ACM Multimedia*, 2001, pp. 411–420.

[8] Xiaodong Wen, Theodore D. Huffmire, Helen H. Hu, and Adam Finkelstein, "Wavelet-based video indexing and querying," vol. 7, no. 5, pp. 350–358, 1999.

[9] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, MIT Press, second edition edition, 2001.

[10] T. Tran Thuong and C. Roisin, *Media content modelling for Authoring and Presenting Multimedia Document*, World Scientific - Series in Machine Perception and Artificial Intelligence, 2002.